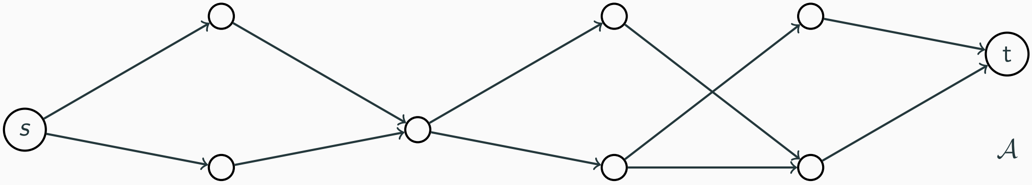
Lower Bounds against Ordered Set-Multilinear ABPs

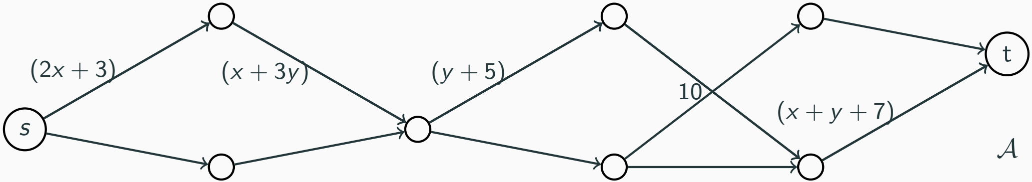
Prerona Chatterjee [with Deepanshu Kush, Shubhangi Saraf and Amir Shpilka]

September 17, 2024

Algebraic Branching Programs

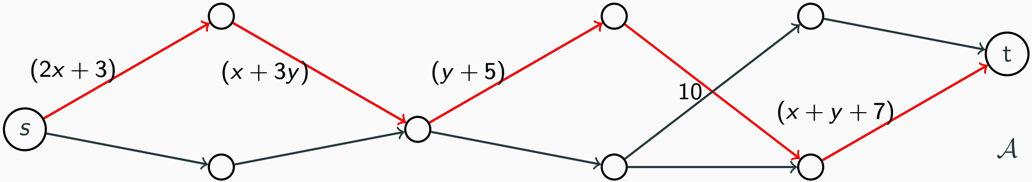


Algebraic Branching Programs



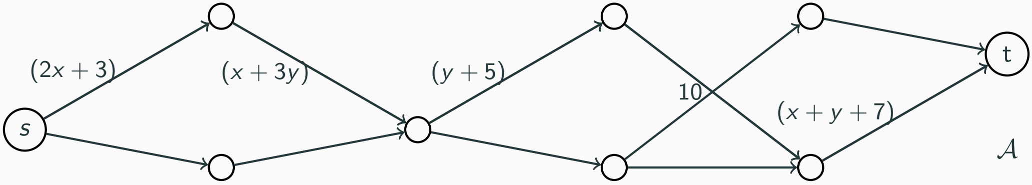
- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$

Algebraic Branching Programs



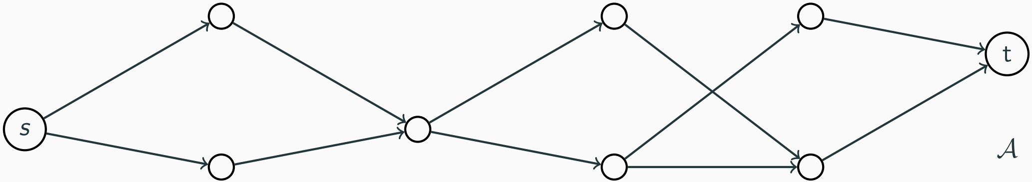
- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$
- Polynomial computed by the path $p = \text{wt}(p)$: Product of the edge labels on p

Algebraic Branching Programs



- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$
- Polynomial computed by the path $p = \mathrm{wt}(p)$: Product of the edge labels on p
- Polynomial computed by the ABP: $f_{\mathcal{A}}(\mathbf{x}) = \sum_p \mathrm{wt}(p)$

Algebraic Branching Programs



- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$
- Polynomial computed by the path $p = \mathrm{wt}(p)$: Product of the edge labels on p
- Polynomial computed by the ABP: $f_{\mathcal{A}}(\mathbf{x}) = \sum_p \mathrm{wt}(p)$

Question: Find explicit polynomials that can not be computed efficiently by ABPs.

Towards Better ABP Lower Bounds

[C-Kumar-She-Volk 22]: Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(nd)$ vertices.

Towards Better ABP Lower Bounds

[C-Kumar-She-Volk 22]: Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(nd)$ vertices.

[Bhargav-Dwivedi-Saxena 24]: Super polynomial lower bound against total-width of $\sum$ osmABP for a polynomial of degree $d = O\left(\frac{\log n}{\log \log n}\right) \implies$ super-polynomial lower bound against ABPs.

Towards Better ABP Lower Bounds

[C-Kumar-She-Volk 22]: Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(nd)$ vertices.

[Bhargav-Dwivedi-Saxena 24]: Super polynomial lower bound against total-width of $\sum$osmABP for a polynomial of degree $d = O\left(\frac{\log n}{\log \log n}\right) \implies$ super-polynomial lower bound against ABPs.

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

- $G_{n,d}$ is computable by a set-multilinear ABP of size $\text{poly}(n)$,
- any $\sum$osmABP computing $G_{n,d}$ must have super-polynomial total-width.

Set-Multilinearity

The variable set is divided into buckets.

$$\mathbf{x} = \mathbf{x}_1 \cup \cdots \cup \mathbf{x}_d \quad \text{where} \quad \mathbf{x}_i = \{x_{i,1}, \ldots x_{i,n_i}\}.$$

Set-Multilinearity

The variable set is divided into buckets.

$$\mathbf{x} = \mathbf{x}_1 \cup \cdots \cup \mathbf{x}_d \quad \text{where} \quad \mathbf{x}_i = \{x_{i,1}, \ldots x_{i,n_i}\}.$$

f is set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

every monomial in f has exactly one variable from $\mathbf{x}_i$ for each $i \in [d]$.

Set-Multilinearity

The variable set is divided into buckets.

$$\mathbf{x} = \mathbf{x}_1 \cup \cdots \cup \mathbf{x}_d \quad \text{where} \quad \mathbf{x}_i = \{x_{i,1}, \ldots x_{i,n_i}\}.$$

f is set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

every monomial in f has exactly one variable from $\mathbf{x}_i$ for each $i \in [d]$.

An ABP is set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if every path in it

computes a set-multilinear monomial with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

Near Tightness of ABP Set-Multilinearisation

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

Near Tightness of ABP Set-Multilinearisation

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

$\sum$osmABP: Sum of ordered set-multilinear ABPs, each with a possibly different ordering.

Near Tightness of ABP Set-Multilinearisation

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

$\sum$osmABP: Sum of ordered set-multilinear ABPs, each with a possibly different ordering.

[B-D-S 24]: Super polynomial lower bound against total-width of $\sum$osmABP for a polynomial of degree $d = O\left(\frac{\log n}{\log \log n}\right) \implies$ super-polynomial lower bound against ABPs.

Near Tightness of ABP Set-Multilinearisation

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

$\sum$osmABP: Sum of ordered set-multilinear ABPs, each with a possibly different ordering.

[B-D-S 24]: Super polynomial lower bound against total-width of $\sum$osmABP for a polynomial of degree $d = O\left(\frac{\log n}{\log \log n}\right) \implies$ super-polynomial lower bound against ABPs.

[C-K-S-S 24]: Super polynomial lower bound against total-width of $\sum$osmABP for a polynomial of degree $d = \omega(\log n)$ that is computable by polynomial-sized ABPs.

Super-Polynomial Lower Bound against $\sum$osmABPs

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

Super-Polynomial Lower Bound against $\sum$osmABPs

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

- $G_{n,d}$ is computable by a set-multilinear ABP of size $\text{poly}(n, d)$,

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

- $G_{n,d}$ is computable by a set-multilinear ABP of size $\text{poly}(n, d)$,
- $G_{n,d}$ can not be computed by a $\sum$ osmABP of total-width $\text{poly}(n)$,

Super-Polynomial Lower Bound against $\sum$osmABPs

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

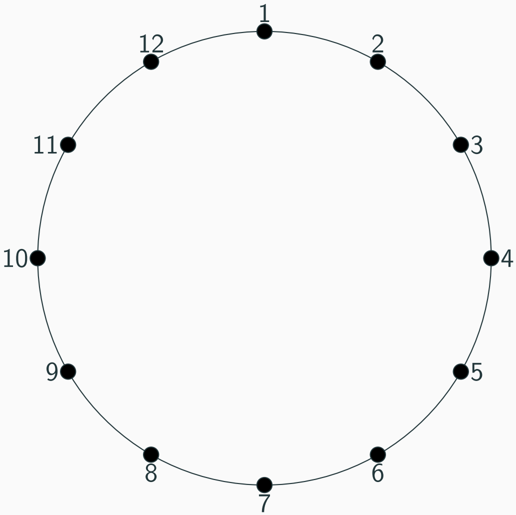
- $G_{n,d}$ is computable by a set-multilinear ABP of size $\text{poly}(n, d)$,
- $G_{n,d}$ can not be computed by a $\sum$osmABP of total-width $\text{poly}(n)$,
- any $\sum$osmABP of max-width $\text{poly}(n)$ computing $G_{n,d}$ requires total-width $2^{\Omega(d)}$,

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

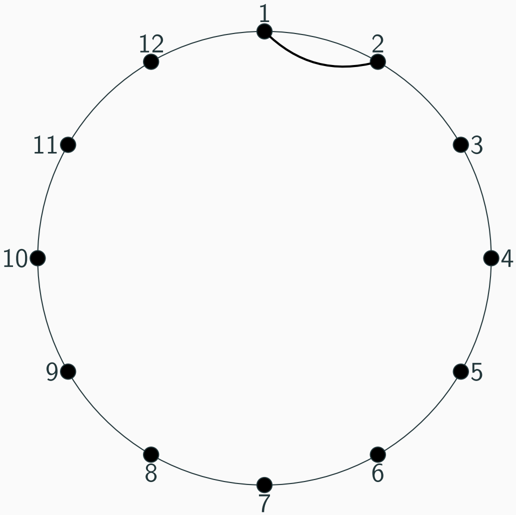
- $G_{n,d}$ is computable by a set-multilinear ABP of size poly(n, d),
- $G_{n,d}$ can not be computed by a $\sum$ osmABP of total-width poly(n),
- any $\sum$ osmABP of max-width poly(n) computing $G_{n,d}$ requires total-width $2^{\Omega(d)}$,
- any ordered set-multilinear branching program computing $G_{n,d}$ requires width $n^{\Omega(d)}$.

Proof Ideas

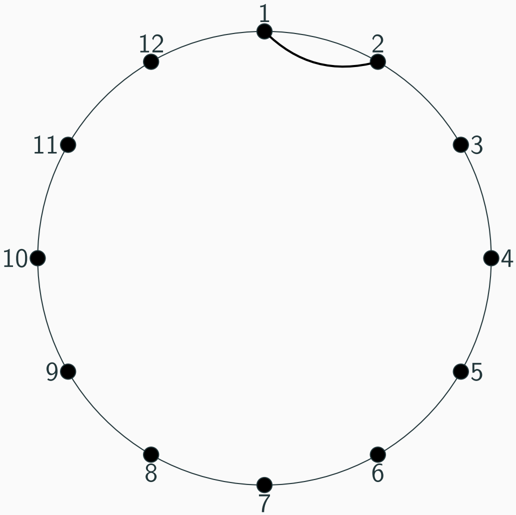
Arc Partition



Arc Partition

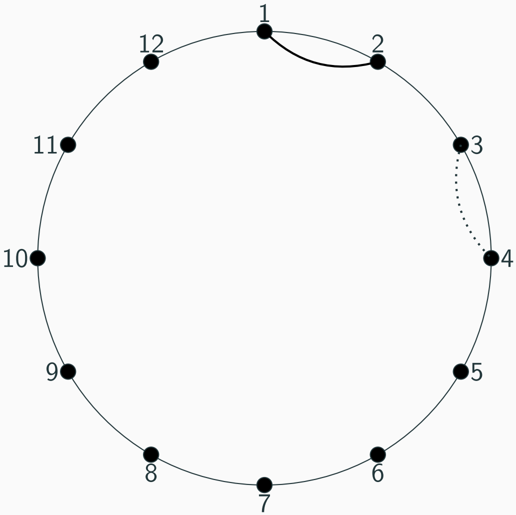


Arc Partition



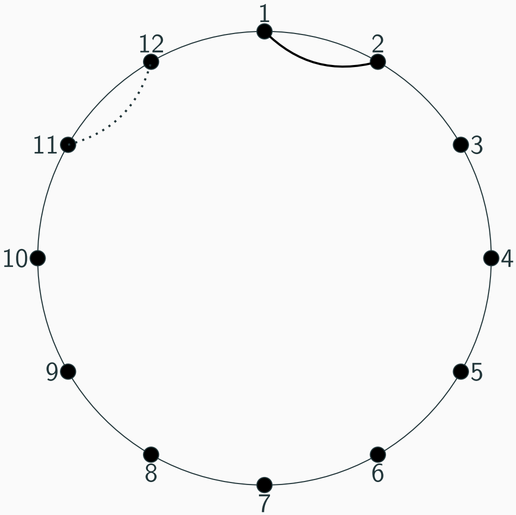
$$\mathcal{P}_1 = \{(1,2)\}$$

Arc Partition



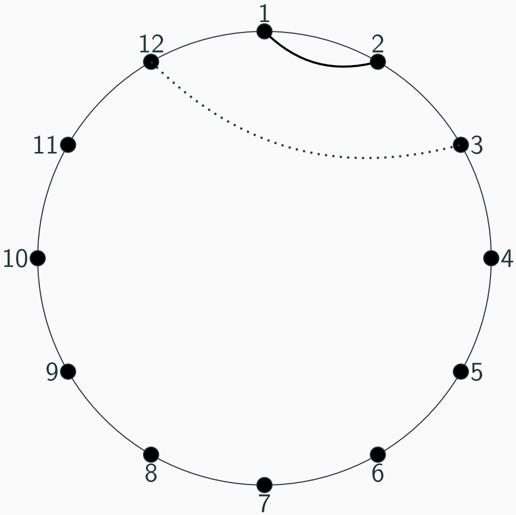
$$\mathcal{P}_1 = \{(1, 2)\}$$

Arc Partition



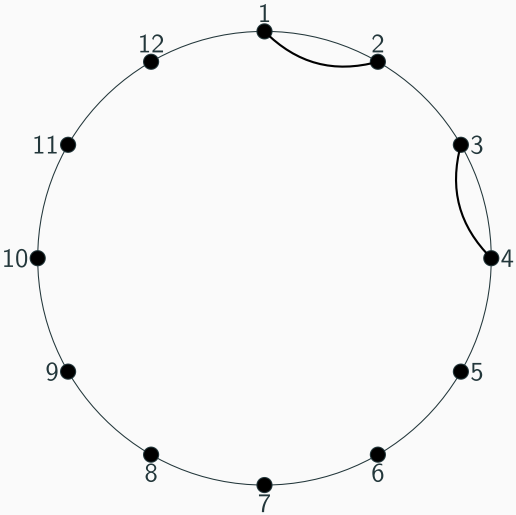
$$\mathcal{P}_1 = \{(1,2)\}$$

Arc Partition



$$\mathcal{P}_1 = \{(1, 2)\}$$

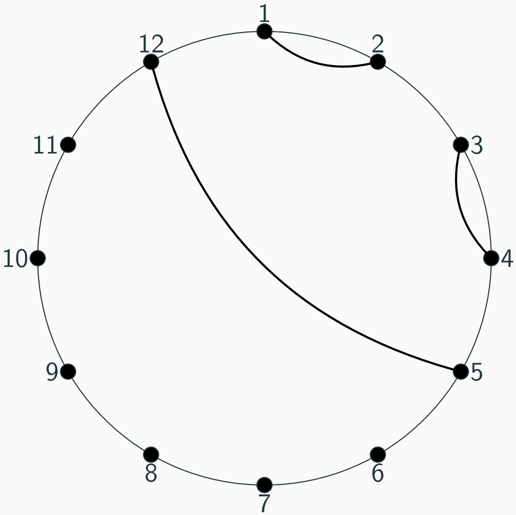
Arc Partition



$$\mathcal{P}_1 = \{(1,2)\}$$

$$\mathcal{P}_2 = \{(1,2),(3,4)\}$$

Arc Partition

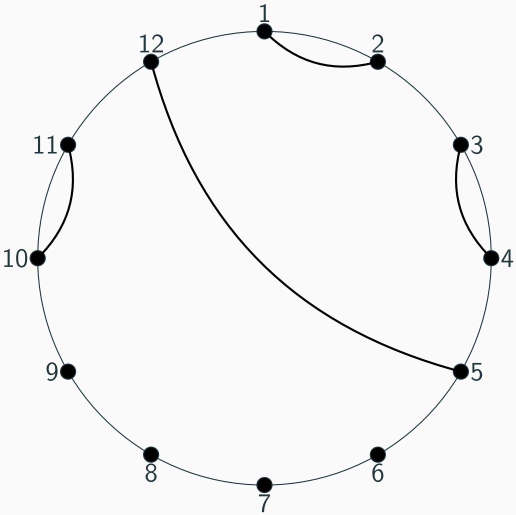


$$\mathcal{P}_1 = \{(1,2)\}$$

$$\mathcal{P}_2 = \{(1,2),(3,4)\}$$

$$\mathcal{P}_3 = \{(1,2),(3,4),(12,5)\}$$

Arc Partition



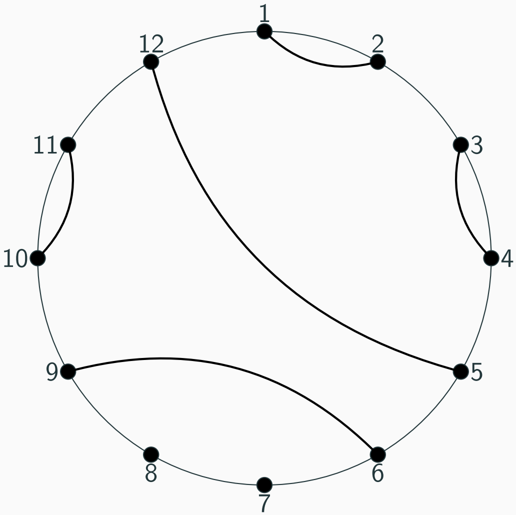
$$\mathcal{P}_1 = \{(1,2)\}$$

$$\mathcal{P}_2 = \{(1,2),(3,4)\}$$

$$\mathcal{P}_3 = \{(1,2),(3,4),(12,5)\}$$

$$\mathcal{P}_4 = \{(1,2),(3,4),(12,5),(10,11)\}$$

Arc Partition



$$\mathcal{P}_1 = \{(1,2)\}$$

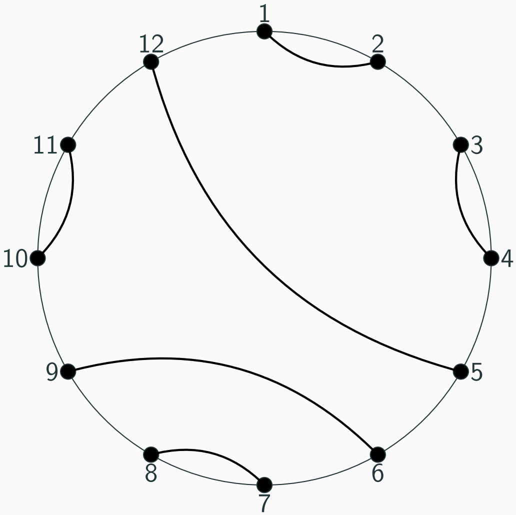
$$\mathcal{P}_2 = \{(1,2),(3,4)\}$$

$$\mathcal{P}_3 = \{(1,2),(3,4),(12,5)\}$$

$$\mathcal{P}_4 = \{(1,2),(3,4),(12,5),(10,11)\}$$

$$\mathcal{P}_5 = \{(1,2),(3,4),(12,5),(10,11),(9,6)\}$$

Arc Partition



$$\mathcal{P}_1 = \{(1, 2)\}$$

$$\mathcal{P}_2 = \{(1, 2), (3, 4)\}$$

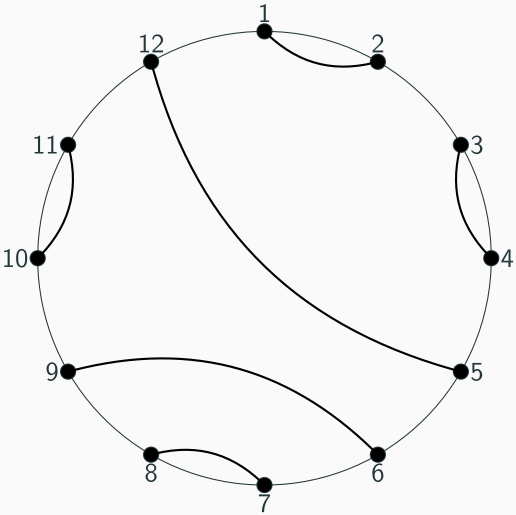
$$\mathcal{P}_3 = \{(1, 2), (3, 4), (12, 5)\}$$

$$\mathcal{P}_4 = \{(1, 2), (3, 4), (12, 5), (10, 11)\}$$

$$\mathcal{P}_5 = \{(1, 2), (3, 4), (12, 5), (10, 11), (9, 6)\}$$

$$\mathcal{P}_6 = \{(1, 2), (3, 4), (12, 5), (10, 11), (9, 6), (8, 7)\}$$

Arc Partition



$$\mathcal{P}_1 = \{(1,2)\}$$

$$\mathcal{P}_2 = \{(1,2),(3,4)\}$$

$$\mathcal{P}_3 = \{(1,2),(3,4),(12,5)\}$$

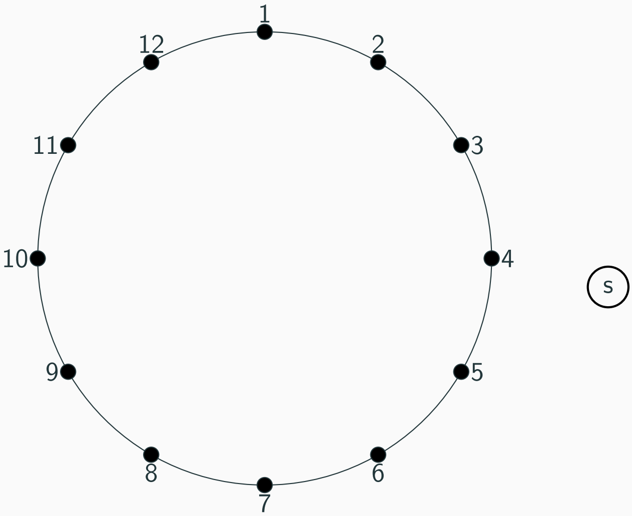
$$\mathcal{P}_4 = \{(1,2),(3,4),(12,5),(10,11)\}$$

$$\mathcal{P}_5 = \{(1,2),(3,4),(12,5),(10,11),(9,6)\}$$

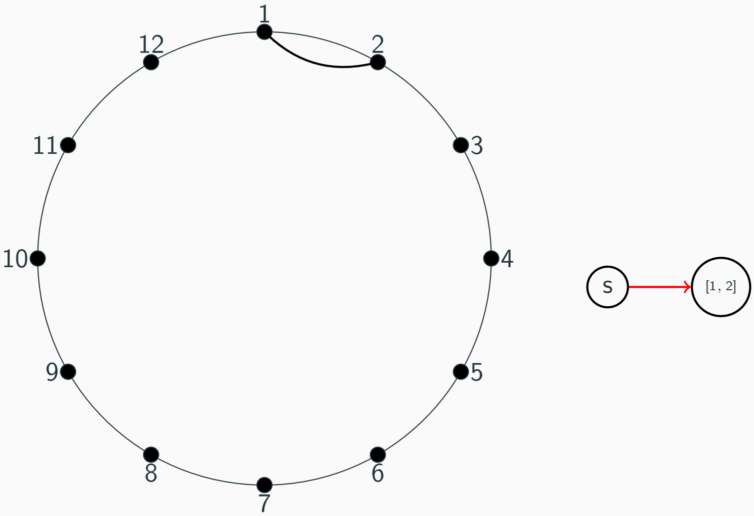
$$\mathcal{P}_6 = \{(1,2),(3,4),(12,5),(10,11),(9,6),(8,7)\}$$

$\mathbf{P}_6 =$ All possibles sequences of such pairs.

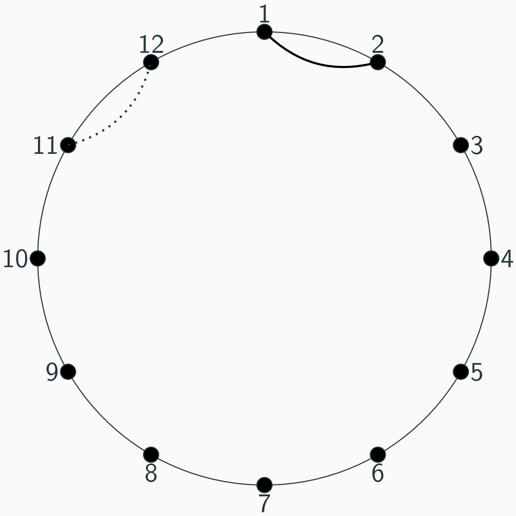
The ABP Upper Bound

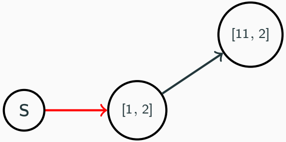


The ABP Upper Bound

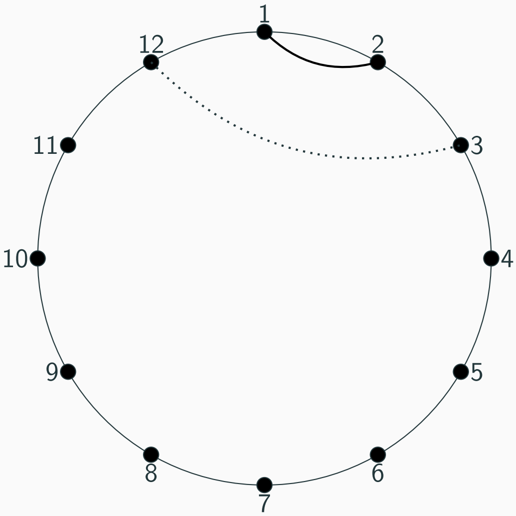


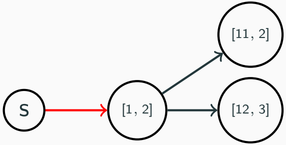
The ABP Upper Bound



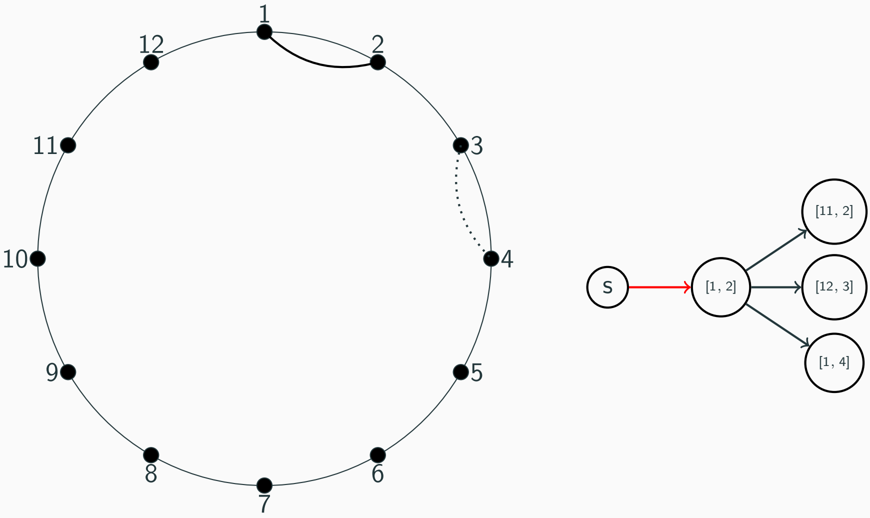


The ABP Upper Bound

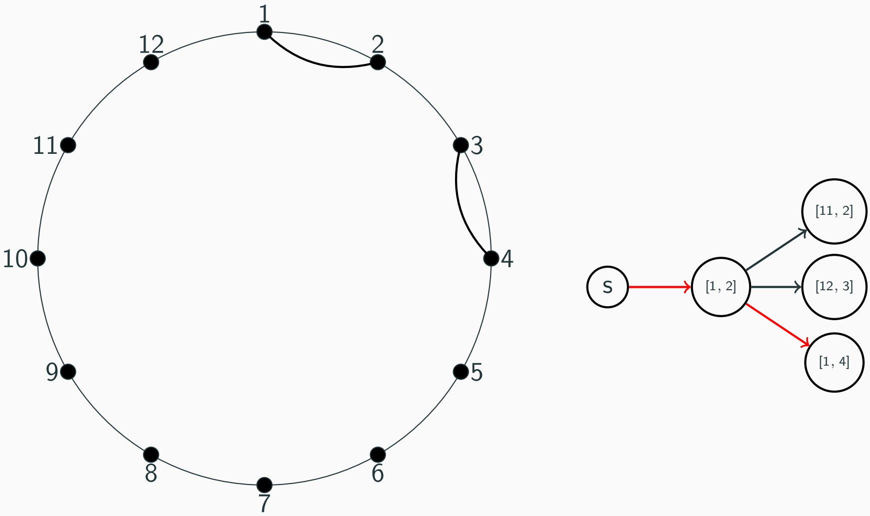




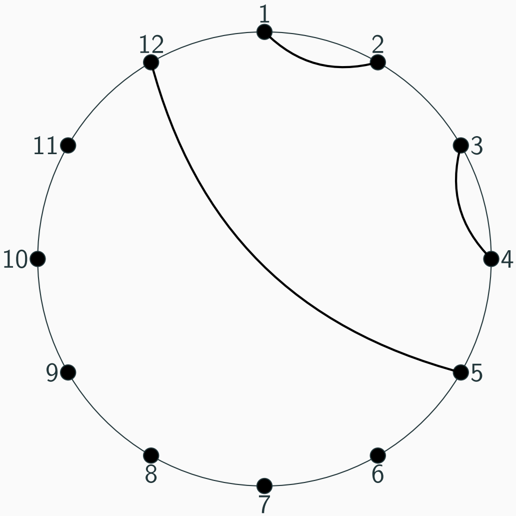
The ABP Upper Bound

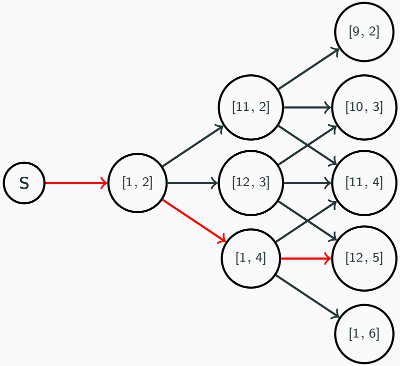


The ABP Upper Bound

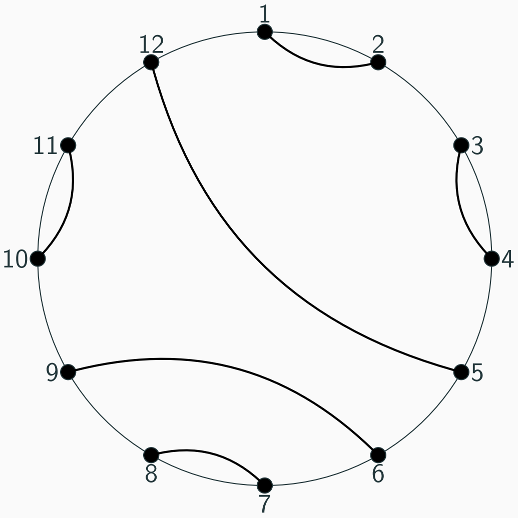


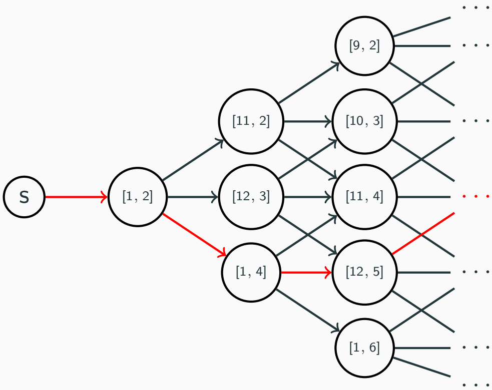
The ABP Upper Bound



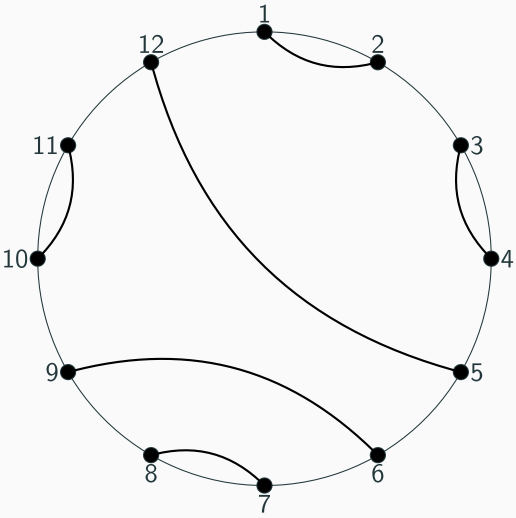


The ABP Upper Bound

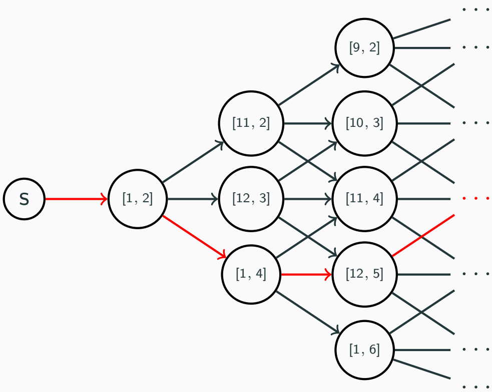




The ABP Upper Bound



Every path corresponds to an element in $\mathbf{P}_{d/2}$.

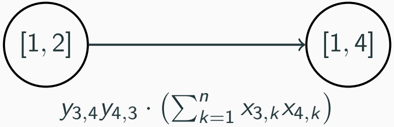


The Hard Polynomial

The Hard Polynomial

The new pair: $(3, 4)$.

The Hard Polynomial



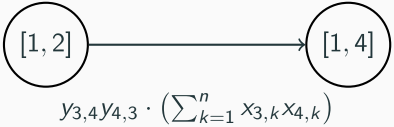
The new pair: $(3, 4)$.

The Hard Polynomial

$y_{3,4}y_{4,3} \cdot \left(\sum_{k=1}^{n} x_{3,k}x_{4,k}\right)$

The new pair: $(3, 4)$.

$(y_{3,4}y_{4,3})$: To select.



The new pair: $(3, 4)$.

$(y_{3,4} y_{4,3})$: To select.

$\left(\sum_{k=1}^{n} x_{3,k} x_{4,k}\right)$: To achieve full-rank.

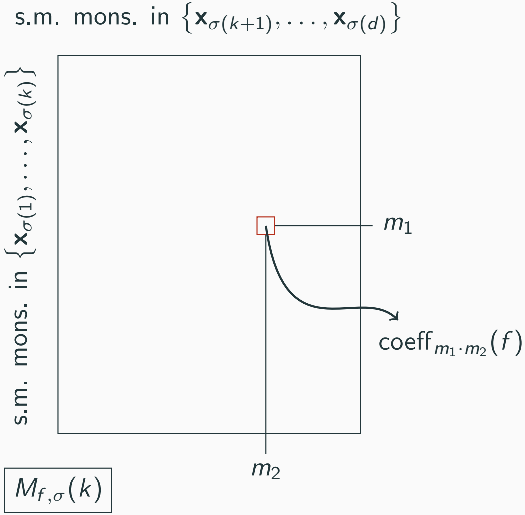
$y_{3,4}y_{4,3} \cdot \left(\sum_{k=1}^{n} x_{3,k}x_{4,k}\right)$

$$
\begin{array}{c|ccccc|}
 & x_{4,1} & x_{4,2} & \cdots & \cdots & x_{4,n} \\
\hline
x_{3,1} & 1 & 0 & \cdots & \cdots & 0 \\
x_{3,2} & 0 & 1 & \cdots & \cdots & 0 \\
\vdots & \vdots & \vdots & & & \vdots \\
\vdots & \vdots & \vdots & & & \vdots \\
x_{3,n} & 0 & 0 & \cdots & \cdots & 1 \\
\hline
\end{array}
$$

The new pair: $(3, 4)$.

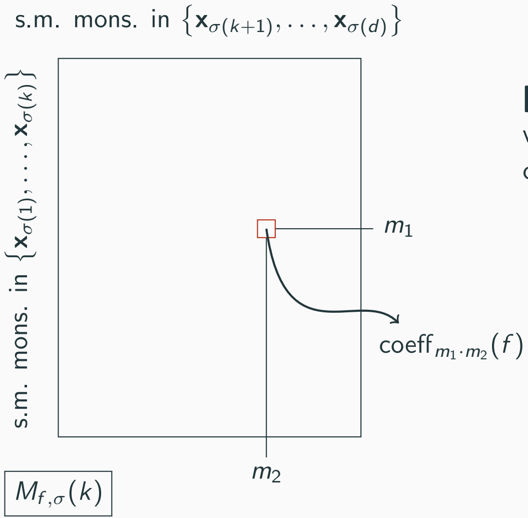
$(y_{3,4}y_{4,3})$: To select.

$\left(\sum_{k=1}^{n} x_{3,k}x_{4,k}\right)$: To achieve full-rank.



f is a set-multilinear poly. w.r.t $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

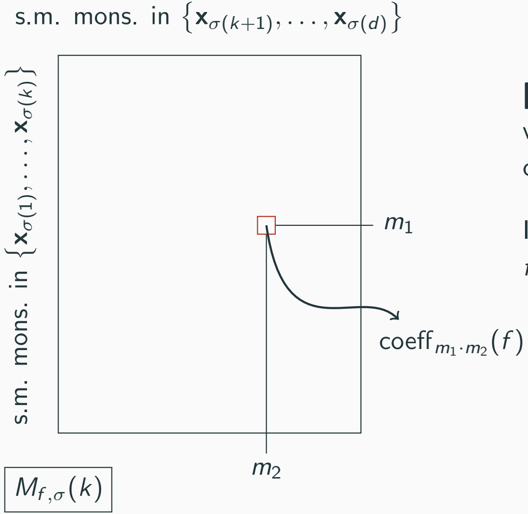
Lower Bound for a single osmABP



f is a set-multilinear poly. w.r.t $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

[Nisan 91]: For every $1 \leq k \leq d$, the number of vertices in the k-th layer of the smallest osmABP(σ) computing f is equal to the rank of $M_{f,\sigma}(k)$.

Lower Bound for a single osmABP



f is a set-multilinear poly. w.r.t $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

[Nisan 91]: For every $1 \leq k \leq d$, the number of vertices in the k-th layer of the smallest osmABP(σ) computing f is equal to the rank of $M_{f,\sigma}(k)$.

If $\mathcal{A}$ is the smallest osmABP (in order σ) computing f, then

$$\mathrm{size}(\mathcal{A}) = \sum_{i=1}^{d} \mathrm{rank}(M_{f,\sigma}(k)).$$

Lower Bound for a single osmABP (contd.)

$$G_{n,d} = \sum_{\mathcal{P} \in \mathbf{P}_{d/2}} \prod_{(i,j) \in \mathcal{P}} y_{i,j} y_{j,i} \cdot \left(\sum_{k=1}^{n} x_{i,k} x_{j,k} \right).$$

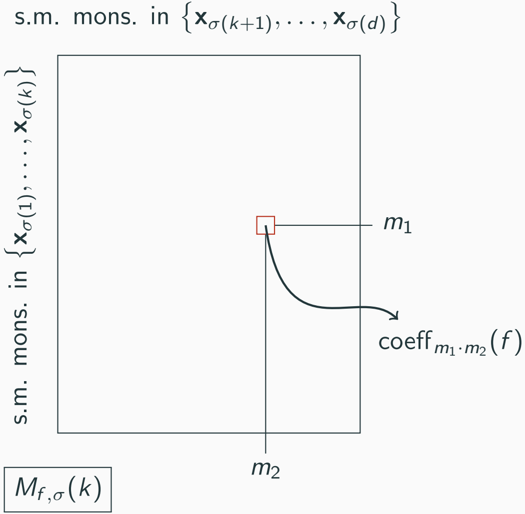
Lower Bound for a single osmABP (contd.)

$$G_{n,d} = \sum_{\mathcal{P} \in \mathbf{P}_{d/2}} \prod_{(i,j) \in \mathcal{P}} y_{i,j} y_{j,i} \cdot \left(\sum_{k=1}^{n} x_{i,k} x_{j,k} \right).$$

Properties:

- $G_{n,d}$ is computable by a set-multilinear ABP of size poly(n, d).

Lower Bound for a single osmABP (contd.)

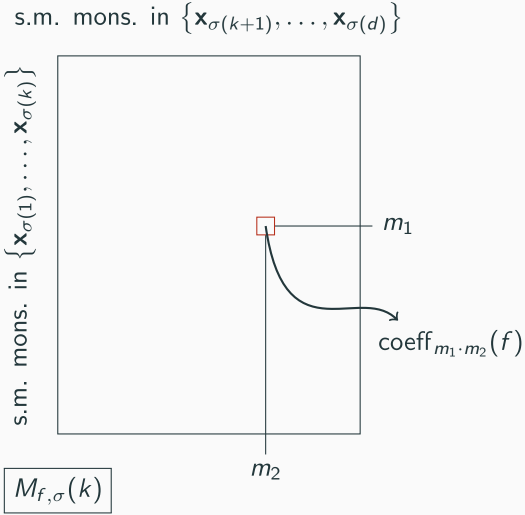


$$G_{n,d} = \sum_{\mathcal{P} \in \mathbf{P}_{d/2}} \prod_{(i,j) \in \mathcal{P}} y_{i,j} y_{j,i} \cdot \left(\sum_{k=1}^{n} x_{i,k} x_{j,k} \right).$$

Properties:

- $G_{n,d}$ is computable by a set-multilinear ABP of size poly(n, d).
- For every $\sigma \in S_d$, there is some $\mathcal{P}$ such that for at least $d/8$ of the $P = (i,j) \in \mathcal{P}$, $i \in \{\sigma(1), \dots \sigma(\frac{d}{2})\}$ & $j \in \{\sigma(1 + \frac{d}{2})), \dots \sigma(d)\}$.

Lower Bound for a single osmABP (contd.)



$$G_{n,d} = \sum_{\mathcal{P} \in \mathbf{P}_{d/2}} \prod_{(i,j) \in \mathcal{P}} y_{i,j} y_{j,i} \cdot \left(\sum_{k=1}^{n} x_{i,k} x_{j,k} \right).$$

Properties:

- $G_{n,d}$ is computable by a set-multilinear ABP of size $\text{poly}(n, d)$.
- For every $\sigma \in S_d$, there is some $\mathcal{P}$ such that for at least $d/8$ of the $P = (i,j) \in \mathcal{P}$, $i \in \{\sigma(1), \ldots \sigma(\frac{d}{2})\}$ & $j \in \{\sigma(1 + \frac{d}{2})), \ldots \sigma(d)\}$.

  Therefore,

$$\text{rank}(M_{G_{n,d},\sigma}(d/2)) = \Omega(n^{d/8}).$$

Lower Bound for a Sum of osmABPs

- $\{M_w(f) \ : \ w \in \mathcal{S}\}$ is a set of matrices such that $M_w(G_{n,d})$ has full rank for every $w \in \mathcal{S}$.

Lower Bound for a Sum of osmABPs

- $\{M_w(f) \ : \ w \in \mathcal{S}\}$ is a set of matrices such that $M_w(G_{n,d})$ has full rank for every $w \in \mathcal{S}$.
- If $G_{n,d}$ is computed by a sum of t osmABPs, then

$$G_{n,d} = \sum_{i=1}^{t} g_i \quad \text{where} \quad g_i = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g^{(i)}_{u_{j-1},u_j}.$$

Lower Bound for a Sum of osmABPs

- $\{M_w(f) \; : \; w \in \mathcal{S}\}$ is a set of matrices such that $M_w(G_{n,d})$ has full rank for every $w \in \mathcal{S}$.
- If $G_{n,d}$ is computed by a sum of t osmABPs, then

$$G_{n,d} = \sum_{i=1}^{t} g_i \quad \text{where} \quad g_i = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g^{(i)}_{u_{j-1},u_j}.$$

- Define a distribution $\mathcal{D}$ on $\mathcal{S}$ such that when $w \sim \mathcal{D}$, if g_is are computable by osmABPs efficiently, then

Lower Bound for a Sum of osmABPs

- $\{M_w(f) \;:\; w \in \mathcal{S}\}$ is a set of matrices such that $M_w(G_{n,d})$ has full rank for every $w \in \mathcal{S}$.
- If $G_{n,d}$ is computed by a sum of t osmABPs, then

$$G_{n,d} = \sum_{i=1}^{t} g_i \quad \text{where} \quad g_i = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g^{(i)}_{u_{j-1},u_j}.$$

- Define a distribution $\mathcal{D}$ on $\mathcal{S}$ such that when $w \sim \mathcal{D}$, if g_is are computable by osmABPs efficiently, then

for every i, w.h.p. there are many js, for which $M_w(g^{(i)}_{u_{j-1},u_j})$ is far from full rank

Lower Bound for a Sum of osmABPs

- $\{M_w(f) \ : \ w \in \mathcal{S}\}$ is a set of matrices such that $M_w(G_{n,d})$ has full rank for every $w \in \mathcal{S}$.
- If $G_{n,d}$ is computed by a sum of t osmABPs, then

$$G_{n,d} = \sum_{i=1}^{t} g_i \quad \text{where} \quad g_i = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g^{(i)}_{u_{j-1},u_j}.$$

- Define a distribution $\mathcal{D}$ on $\mathcal{S}$ such that when $w \sim \mathcal{D}$, if g_is are computable by osmABPs efficiently, then

for every i, w.h.p. there are many js, for which $M_w(g^{(i)}_{u_{j-1},u_j})$ is far from full rank

$\implies$ for every i, w.h.p. $M_w(g_i)$ is far from full rank

Lower Bound for a Sum of osmABPs

- $\{M_w(f) \ : \ w \in \mathcal{S}\}$ is a set of matrices such that $M_w(G_{n,d})$ has full rank for every $w \in \mathcal{S}$.
- If $G_{n,d}$ is computed by a sum of t osmABPs, then

$$G_{n,d} = \sum_{i=1}^{t} g_i \quad \text{where} \quad g_i = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g^{(i)}_{u_{j-1},u_j}.$$

- Define a distribution $\mathcal{D}$ on $\mathcal{S}$ such that when $w \sim \mathcal{D}$, if g_is are computable by osmABPs efficiently, then

for every i, w.h.p. there are many js, for which $M_w(g^{(i)}_{u_{j-1},u_j})$ is far from full rank

$\implies$ for every i, w.h.p. $M_w(g_i)$ is far from full rank

$\implies$ $M_w(G_{n,d})$ is far from full rank unless t is large.

Open Questions

- What happens when $\omega\left(\frac{\log n}{\log \log n}\right) \leq d \leq O(\log n)$?

Open Questions

- What happens when $\omega\left(\frac{\log n}{\log\log n}\right) \leq d \leq O(\log n)$?
- Super-quadratic lower bound against set-multilinear *ABPs*.

Open Questions

- What happens when $\omega\left(\frac{\log n}{\log\log n}\right) \leq d \leq O(\log n)$?
- Super-quadratic lower bound against set-multilinear *ABPs*.

Thank you!!!

Discussion

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log \log n)}$.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log \log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log \log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.
- Separation is not tight: $\mathrm{IMM}_{n,n}$ is computable by a set-multilinear formula of size $n^{O(\log n)}$.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log \log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.
- Separation is not tight: $\mathrm{IMM}_{n,n}$ is computable by a set-multilinear formula of size $n^{O(\log n)}$.

Remarks

- Set-multilinear formula LBs imply formula LBs in low degree.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log\log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.
- Separation is not tight: $\mathrm{IMM}_{n,n}$ is computable by a set-multilinear formula of size $n^{O(\log n)}$.

Remarks

- Set-multilinear formula LBs imply formula LBs in low degree.

  [Raz]: For $d = O(\frac{\log n}{\log\log n})$, if f is computable by a formula of size $s = \mathrm{poly}(n)$, then it is also computable by a set-multilinear formula of size $\mathrm{poly}(n)$.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log \log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.
- Separation is not tight: $\mathrm{IMM}_{n,n}$ is computable by a set-multilinear formula of size $n^{O(\log n)}$.

Remarks

- Set-multilinear formula LBs imply formula LBs in low degree.

  [Raz]: For $d = O(\frac{\log n}{\log \log n})$, if f is computable by a formula of size $s = \mathrm{poly}(n)$, then it is also computable by a set-multilinear formula of size $\mathrm{poly}(n)$.
- $n^{\Omega(\log n)}$ set-multilinear formula LB for $\mathrm{IMM}_{n,n}$ implies formula LB due to self-reducibility.

[Kush-Saraf]

The Result: There is an n-variate polynomial, $G_{n,n}$, of degree n computable by a set-multilinear ABP of width $s = \text{poly}(n)$ such that any set-multilinear formula computing it has size $n^{\Omega(\log n)}$.

[Kush-Saraf]

The Result: There is an n-variate polynomial, $G_{n,n}$, of degree n computable by a set-multilinear ABP of width $s = \text{poly}(n)$ such that any set-multilinear formula computing it has size $n^{\Omega(\log n)}$.

Wait! Why doesn't that lead to formula lower bounds?

[Kush-Saraf]

The Result: There is an n-variate polynomial, $G_{n,n}$, of degree n computable by a set-multilinear ABP of width $s = \text{poly}(n)$ such that any set-multilinear formula computing it has size $n^{\Omega(\log n)}$.

Wait! Why doesn't that lead to formula lower bounds?

- f is a projection of $\mathrm{IMM}_{s,n}$
- $\mathrm{IMM}_{s,n}$ is self-reducible

[Kush-Saraf]

The Result: There is an n-variate polynomial, $G_{n,n}$, of degree n computable by a set-multilinear ABP of width $s = \text{poly}(n)$ such that any set-multilinear formula computing it has size $n^{\Omega(\log n)}$.

Wait! Why doesn't that lead to formula lower bounds?

- f is a projection of $\mathrm{IMM}_{s,n}$
- $\mathrm{IMM}_{s,n}$ is self-reducible

But f need not be self-reducible unless f is computable by an osmABP.

[Kush-Saraf]

The Result: There is an n-variate polynomial, $G_{n,n}$, of degree n computable by a set-multilinear ABP of width $s = \text{poly}(n)$ such that any set-multilinear formula computing it has size $n^{\Omega(\log n)}$.

Wait! Why doesn't that lead to formula lower bounds?

- f is a projection of $\mathrm{IMM}_{s,n}$
- $\mathrm{IMM}_{s,n}$ is self-reducible

But f need not be self-reducible unless f is computable by an osmABP.

How different are the powers of smABP**s and** osmABP**s?**

What Our Results Say

- Optimal separation between smABP and osmABP.
- Exponential separation between smABP and $\sum$ osmABP when $d = \Theta(n)$.
- Super-polynomial separation between smABP and $\sum$ osmABP when $d = \omega(n)$.

Thank you again!!!